IN THE CLAIMS:

1. (Currently amended) A method in a data processing system for accessing a client service, the method comprising:

managing a pool of connections to [[the]] client service instances;

responsive to a request from a ~~user application~~ client from a plurality of ~~user applications~~ clients, assigning a connection to a client service instance to the client ~~service~~ from the pool of connections to the client service instances;

invoking the request on the client using the connection to the client service instance; [[and]]

responsive to receiving a response to the request from the client service instance, returning the result to the client ~~user application~~; and

placing the request in a queue if there is no free client service instance within the pool of connections to the client service instances.

2. (Currently amended) The method of claim 1 further comprising:

freeing the client service instance back to the pool after invoking the request on the client service.

3. (Currently amended) The method of claim 1 further comprising:

waiting for the response from the client service instance after the client service instance has been invoked; and

responsive to a timeout occurring while waiting for the response, returning a response to the [[user]] client indicating that the timeout has occurred.

4. (Currently amended) The method of claim 1, wherein the ~~user application~~ client is a ~~client~~ user application.

5. (Currently amended) The method of claim 1, wherein the client service instance is an application programming interface to a server process.

6. (Original) The method of claim 5, wherein the server process is located on a remote data processing system.

7. (Currently amended) The method of claim 1, wherein the pool of connections to the client service instances ~~services~~ is used to access report services on a server.

8. (Currently amended) The method of claim 1, wherein the ~~response~~ result is returned immediately upon receiving the response from the client service.

9. (Currently amended) The method of claim 1, wherein [[a]] an error message is returned to the ~~user application~~ client after a period of time passes without receiving the response.

10. (Canceled)

11. (Original) The method of claim 1, wherein a particular client service instance only accepts and processes one request at a time.

12. (Currently amended) The method of claim [[10]] 7, wherein the server service is located on a remote data processing system.

13. (Original) A method in a data processing system for accessing a client service, the method comprising:

receiving requests for the client service, wherein the client service is a single-threaded process;

queuing a new request if a current request has been invoked on the client service;

responsive to receiving a response to the current request from the client service, returning the result to a requestor of the current request; and

invoking the new request on the client service.

14. (Original) The method of claim 13, wherein requests are sent to the client service form the queue in a first-in-first-out basis.

15. (Original) The method of claim 13, wherein the client service is used to access a server process in a server.

16. (Original) The method of claim 13, wherein the client service is an application programming interface to a server process.

17. (Currently amended) A data processing system comprising:

a bus system;

a communications unit connected to the bus system;

a memory connected to the bus system, wherein the memory includes as set of instructions; and

a processing unit connected to the bus system, wherein the processing unit executes the set of instructions to manage a pool of connections to the client service instances; assign a connection to a client service instance to the client from the pool of connections to the client service instances in response to a request from a client from a plurality of clients; invoke the request on the a client ~~service~~ using the connection to the client service instance; [[and]] return the result to the [[user]] client in response to receiving a response from the client service; and place the request in a queue if there is no free client service instance within the pool of connections to the client service instances.

18. (Original) A data processing system comprising:

a bus system;

a communications unit connected to the bus system;

a memory connected to the bus system, wherein the memory includes as set of instructions; and

a processing unit connected to the bus system, wherein the processing unit executes the set of instructions to receive requests for the client service, wherein the client service is a single-threaded process; queue a new request if a current request has been invoked on the client service; return the result to a requestor of the current request in response to receiving a response to the current request from the client service; and invoke the new request on the client service.

19. (Currently amended) A data processing system for accessing a client service, the data processing system comprising:

managing means for managing a pool of connections to [[the]] client service instances;

assigning means, responsive to a request from a ~~user application~~ client from a plurality of ~~user applications~~ clients, for assigning a connection to a client service instance to the client ~~service~~ from the pool of connections to the client service instances;

invoking means for invoking the request on the client using the connection to the client service instance; [[and]]

returning means, responsive to receiving a response from the client service, for returning the result to the ~~user application~~ client; and

placing means for placing the request in a queue if there is no free client service instance within the pool of connections to the client service instances.

20. (Currently amended) The data processing system of claim 19 further comprising:

freeing means for freeing the client service instance back to the pool after invoking the request on the client service.

21. (Currently amended) The data processing system of claim 19 further comprising:

waiting means for waiting for the response from the client service after the client service has been invoked; and

responsive to a timeout occurring while waiting for the response, returning a response to the [[user]] client indicating that the timeout has occurred.

22. (Currently amended) The data processing system of claim 19, wherein the ~~user application~~ client is a ~~client~~ user application.

23. (Currently amended) The data processing system of claim 19, wherein the client service instancc is an application programming interface to a server process.

24. (Original) Thc data processing system of claim 23, wherein the server process is located on a remote data processing system.

25. (Currently amended) The data processing system of claim 19, wherein the pool of connections to the client service instances ~~services~~ is used to access report services on a servcr.

26. (Currently amended) The data processing system of claim 19, wherein the ~~response~~ result is returned immediately upon receiving the responsc from the client scrvice.

27. (Currcntly amended) The data processing system of claim 19, wherein [[a]] an error message is returned to the ~~user application~~ client after a period of time passes without recciving the response.

28. (Canceled)

29. (Original) The data processing system of claim 19, wherein a particular client service instance only accepts and processes one request at a time.

30. (Currently amended) The data processing system of claim [[27]] 25, wherein thc server service is located on a remote data processing system.

31. (Original) A data processing system for accessing a client service, the data processing system comprising:

receiving means for receiving rcquests for the client service, wherein the client service is a singlc-threaded process;

queuing means for queuing a new request if a current request has been invoked on the client service;

returning means, responsive to receiving a response to the current request from the client service, for returning the result to a requestor of the current request; and

invoking means for invoking the new request on the client service.

32. (Original) The data processing system of claim 31, wherein requests are sent to the client service form the queue in a first-in-first-out basis.

33. (Original) The data processing system of claim [[30]] 31, wherein the client service is used to access a server process in a server.

34. (Original) The data processing system of claim [[30]] 31, wherein the client service is an application programming interface to a server process.

35. (Currently amended) A computer program product in a computer readable medium for accessing a client service, the computer program product comprising:

first instructions for managing a pool of connections to [[the]] client service instances;

second instructions, responsive to a request from a ~~user application~~ client from a plurality of ~~user applications~~ clients, for assigning a connection to a client service instance to the client ~~service~~ from the pool of connections to the client service instances;

third instructions for invoking the request on the client using the connection to the client service instance; [[and]]

fourth instructions, responsive to receiving a response to the request from the client service instance, for returning the result to the ~~user application~~ client; and

fifth instructions for placing the request in a queue if there is no free client service instance within the pool of connections to the client service instances.

36. (Original) A computer program product in a computer readable medium for accessing a client service, the computer program product comprising:

first instructions for receiving requests for the client service, wherein the client service is a single-threaded process;

second instructions for queuing a new request if a current request has been invoked on the client service;

third instructions, responsive to receiving a response to the current request from the client service, for returning the result to a requestor of the current request; and

fourth instructions for invoking the new request on the client service.